CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1-31.

- After this Amendment: Claims 1-7, 9, and 12-31.

Non-Elected, Canceled, or Withdrawn claims: 8, 10, and 11.

Amended claims: 1, 3, 4, 9, 13, 14, and 19.

New claims: None.

Claims:

1. (Proposed Amended)   A method for generating an installation file for a particular version of a relational database, the method comprising:

given a database system of a particular version, the particular version being unknown prior to execution of the method, determining the particular version of the relational database wherein metadata exists that describes a sequence of multiple versions of the relational database where each version is an upgrade from a previous version, and the particular version is not a first version in the sequence;

automatically determining a first set of scripts, the first set of scripts comprising of data definition language (DDL) scripts associated with implementing the particular version of the relational database, wherein the first set of scripts are selected from a plurality of scripts, one or more of the plurality of scripts being not associated with implementing the particular version;

the automatically determining a first set of scripts comprising:

extracting a set A₁ comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version;

iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an ith version of the relational database, the zero or more filenames each associated with a file comprising a data definition language script executed when upgrading from version i-1 of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j;

determining the first set of scripts as a set theory union of sets A_1, $A_2, ..., A_j$;

automatically determining a second set of scripts, the second set of scripts comprising ~~of~~ data manipulation language scripts associated with implementing the particular version of the relational database, wherein the second set of scripts are selected from a plurality of scripts, one or more of the plurality of scripts being not associated with implementing the particular version, the determining a second set of scripts comprising:

extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an ith version of the relational database, the zero or more filenames each associated with a file comprising a DML script

to be executed to add or modify a DML object when upgrading from version i-1 of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j;

iteratively extracting a set B_i comprising zero or more filenames from metadata associated with an ith version of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version i-1 of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j; and determining the second set C_j by determining:
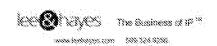
$$C_2 = [A_1 \cup A_2] - B_2,$$
$$C_3 = [C_2 \cup A_3] - B_3,$$
$$C_4 = [C_3 \cup A_4] - B_4$$

$$C_j = [C_{j-1} \cup A_j] - B_j \text{ ;and}$$

generating an installation file comprising a set theory union of the first set of scripts and the second set of scripts.

2. (Original)   The method as recited in claim 1 wherein the particular version is associated with a first version in a sequence of one or more versions of the relational database.

3. (Currently Amended)   The method as recited in claim 1 wherein the automatically determining a first set comprises extracting a filename from metadata associated with the particular version ~~version~~, the filename associated with a file comprising a data definition language script.

4. (Currently Amended) The method as recited in claim 1 wherein the automatically determining a second set comprises extracting a filename from metadata associated with the particular version ~~version~~, the filename associated with a file comprising a data manipulation language script.

5. (Previously Presented) The method as recited in claim 1 wherein the generating an installation file comprises copying a data definition language script from a script file associated with the first set into the installation file.

6.   (Previously Presented) The method as recited in claim 1 wherein the generating an installation file comprises copying a data manipulation language script from a script file associated with the second set into the installation file.

7. (Previously Presented) The method as recited in claim 6 wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file.

8. (Canceled)

9. (Currently Amended) The method as recited in claim ~~8~~ _1_ wherein the metadata comprises an XML file.

10. (Canceled)

11. (Canceled)

12. (Previously Presented) One or more computer-readable media having computer-readable instructions recorded thereon which, when executed by a computer, cause the computer to implement the method as recited in claim 1.

13. (Currently Amended) A method for generating an upgrade file to upgrade version i of a relational database to version j of the relational database, where $j > i$, the method comprising:

determining a set A of data definition language (DDL) scripts that, when executed, perform creates, alters, and drops of DDL objects associated with version i of the

relational database, resulting in DDL objects associated with version *j* of the relational database;

determining a set B of data manipulation language (DML) scripts that, when executed, create DML objects that are associated with version *j* of the relational database, but that are not associated with version *i* of the relational database;

determining a set C of DML scripts that, when executed, modify DML objects that are associated with both version *i* and version *j* of the relational database, but that differ between version *i* and version *j* of the relational database;

determining a set D of DML drop scripts that, when executed, drop DML objects that are associated with version *i* of the relational database, but that are not associated with version *j* of the relational database; and

generating an upgrade file comprising a ~~set theory union of sets~~ union by set ~~theory of sets~~ A, B, C, and D (A $\bigcup$ B $\bigcup$ C $\bigcup$ D).~~;~~

14. (Currently Amended) The method as recited in claim 13 wherein the determining a set A comprises:

iteratively extracting sets M_k, each comprising zero or more filenames from metadata associated with a kth version of the relational database, where $i < k <= j$,~~,~~ the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version *k*-1 of the relational database to version *k* of the relational database; and
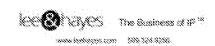
determining the set A as the set theory union of sets ~~union by set theory of sets~~ $M_{i+1}, M_{i+2}, ..., M_j$ ($A = M_{i+1} \bigcup M_{i+2} \bigcup ... \bigcup M_j$).

15. (Original) The method as recited in claim 13 wherein the determining a set B comprises:

determining a set E of DML scripts that when executed:

perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and

perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database; and

determining set B as the difference between sets E and C ($B = E - C$).

16. (Original) The method as recited in claim 15 wherein the determining a set E comprises:

iteratively determining a set P_x of DML scripts that when executed will upgrade DML objects from version x-1 of the relational database to version x of the relational database, where x varies incrementally from i+1 to j;

iteratively determining a set N_x of DML scripts that when executed will drop DML objects that are associated with version x-1 of the relational database but that are not associated with version x of the relational database, where x varies incrementally from i+2 to j;

iteratively determining a set M_x of DML scripts that when executed will upgrade DML objects from version i of the relational database to version x of the relational database, where x varies incrementally from i+1 to j, and where:

$$M_{i+1} = P_{i+1}$$

$$M_{i+2} = [M_{i+1} \cup P_{i+2}] - N_{i+2}$$

$$M_{i+3} = [M_{i+2} \cup P_{i+3}] - N_{i+3}$$

$$\ldots$$

$$M_j = [M_{j-1} \cup P_j] - N_j; \text{ and}$$

determining set $E = M_j$.

17. (Original)   The method as recited in claim 13 wherein the determining a set C comprises:

determining a set E of DML scripts that when executed:

perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and

perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database;

determining a set F_j of DML scripts that when executed, create DML objects associated with version j of the relational database; and

determining set C as the intersection of set E and set F_j ($C = E \cap F_j$).

18. (Original) The method as recited in claim 17 wherein the determining a set F_j comprises:

extracting a set M_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

iteratively extracting a set M_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version x-1 of the relational database to version x of the relational database, where x varies incrementally from 2 to j;

iteratively extracting a set B_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version x-1 of the relational database to version x of the relational database, where x varies incrementally from 2 to j; and

determining the set F_j by determining:
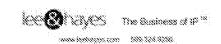
$$F_2 = [M_1 \bigcup M_2] - B_2,$$
$$F_3 = [F_2 \bigcup M_3] - B_3,$$
$$F_4 = [F_3 \bigcup M_4] - B_4,$$

$$\cdots$$
$$F_j = [F_{j-1} \bigcup M_j] - B_j.$$

19. (Currently Amended) The method as recited in claim 13 wherein the determining a set D comprises:

determining a set E of DML scripts that when executed:

perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and

perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database;

iteratively determining a set Fx of DML scripts that when executed, drop DML objects associated with version x-1 of the relational database that are not associated with version x of the relational database, where x varies incrementally from i+1 to j;

determining a set G as the ~~set theory union of sets~~ ~~union by set theory of sets~~ F_i, F_{i+1}, F_{i+2}, ..., F_j ($G = F_i \bigcup F_{i+1} \bigcup ... \bigcup F_{i+2}$); and

determining set D as the difference between set G and set E (D = G - E).

20. (Previously Presented) A system comprising:

a processor;

a memory;

one or more data definition language (DDL) scripts, each associated with one or more versions of a relational database;

one or more data manipulation language (DML) scripts, each associated with one or more versions of the relational database;

a database schema version management structure definition;

schema data associated with multiple versions of the relational database, the schema data organized according to the database schema version management structure definition; and

an installation file generator stored in the memory and executed on the processor to apply laws of set theory to the schema data to generate a file comprising the one or more DDL scripts associated with a particular one of the multiple versions of the relational database, and the one or more DML scripts associated with the particular one of the multiple versions of the relational database.

21. (Original) The system as recited in claim 20 wherein the database schema version management structure definition comprises an XML schema definition.

22. (Original) The system as recited in claim 21 wherein the schema data is maintained in an XML file structured according to the XML schema definition.

23. (Previously Presented) A system comprising:

a memory;

a processor; and

a database schema version management system stored in the memory, and executed on the processor to:

manage schema data associated with multiple versions of a relational database; and

generate an installation file associated with any one of the multiple versions of the relational database.

24. (Original) The system as recited in claim 23 wherein the schema data identifies a script associated with a data definition language object of the relational database.

25. (Original) The system as recited in claim 23 wherein the schema data identifies a script associated with a data manipulation language object of the relational database.

26. (Original) The system as recited in claim 23 wherein the database schema version management system is further configured to generate an installation file associated with an initial version of the relational database.

27. (Original) The system as recited in claim 23 wherein the database schema version management system is further configured to generate an installation file associated with a non-initial version of the relational database.

28. (Previously Presented) The system as recited in claim 23 wherein the database schema version management system is further configured to generate an

upgrade file for upgrading one version of the relational database to another version of the relational database.

29. (Previously Presented) One or more computer-readable media comprising computer-readable instructions recorded thereon, which, when executed, cause a computer system to:

maintain schema data that identifies scripts associated with database objects of multiple sequential versions of a relational database; and

generate an installation file associated with an initial version of the relational database by applying laws of set theory to the schema data to identify scripts associated with the database objects of the initial version of the relational database.

30. (Previously Presented) The one or more computer-readable media as recited in claim 29, further comprising computer-readable instructions recorded thereon, which, when executed, cause a computer system to:

generate an installation file associated with a non-initial version of the relational database by applying laws of set theory to the schema data to identify:

scripts associated with data definition language (DDL) objects that are associated with the non-initial version of the relational database; and

scripts associated with data manipulation language (DML) objects that are associated with the non-initial version of the relational database.

31. (Previously Presented) The one or more computer-readable media as recited in claim 29, further comprising computer-readable instructions recorded thereon, which, when executed, cause a computer system to:

generate an upgrade file associated with an upgrade from a first, but not necessarily initial, version of the relational database to a second, later, but not necessarily immediately sequential, version of the relational database by applying laws of set theory to the schema data to identify:

data definition language (DDL) scripts associated with DDL objects of the relational database that have been created or modified between the first and second versions of the relational database;

data manipulation language (DML) scripts associated with DML objects of the relational database that have been created between the first and second versions of the relational database;

DML scripts associated with DML objects of the relational database that have been modified between the first and second versions of the relational database; and

drop scripts associated with database objects that have been dropped and not re-created between the first and second versions of the relational database.